AN EXPLORATION OF MULTIMODAL DOCUMENT CLASSIFICATION STRATEGIES

BY

SCOTT DEEANN CHEN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Professor Pierre Moulin

ABSTRACT

This thesis explores multimodal document classification algorithms in a unified framework. Classification algorithms are designed to exploit both text and image information, which proliferates in modern documents. We design meta-classification schemes that combine and integrate state-of-the-art text and image feature-extractors with state-of-the-art classifiers. Meta-classifiers fuse information across modalities that differ in nature and hence have more information on hand to make decisions. This thesis also discusses strategies that exploit correlations not only within a single modality but also among modalities. Techniques that exploit correlations within a modality include image meta-feature vector combination and latent Dirichlet allocation-based image meta-feature extraction. Another technique that exploits correlations between text and image cleans image with text information. Experiments on real-world databases from Wikipedia demonstrate the benefits of meta-classification for multimodal documents.

To my family.

ACKNOWLEDGMENTS

I deeply thank my adviser, Professor Pierre Moulin, for his advice and guidance. My thanks also go to Professor Vishal Monga at Pennsylvania State University (formerly at Xerox) for suggesting this project and providing invaluable input. I would like to thank my family, especially my mother, for their endless love and support. I would also like to thank my friends for their encouragement and helpful discussions. Last but not least, I want to thank Ying-Yu Chen for her constant support and great encouragement.

I gratefully acknowledge funding from the Xerox Cooperation and the KLA-Tencor corporation.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Document classification has been actively researched since the 1990s. Document classification applications include database management [1], topic spotting [2], e-mail spam filtering [3], and web-page classification [4].

A classification system is composed of two parts, a feature-extractor and a classifier. A feature-extractor computes features that are relevant for the classification task. A classifier observes the features and makes predictions. Traditionally, document classification has been mainly based on text features. Many text feature-extractors are available in the literature [5] [6] [7], and the most popular ones are variants of term-frequency vectors [8] [9] [10]. A wide variety of data mining, machine learning, and pattern recognition techniques have been applied to document classification. These include naive Bayes classifiers, k-nearest neighbor classifiers, neural networks, decision trees, logistic regression, and support vector machines (SVM) [10].

An increasingly large number of document collections, however, contain not only text but also multimedia data such as images, audio, or video. Examples include official annual reports, advertisement brochures, technical and scientific articles, audio books, news websites, and organized web-page collections such as Wikipedia. Multimedia data in documents often show high correlation with text content. For example, a news web page about an event often includes video of the event (usually accompanied with audio), pictures of the event, and, of course, text describing the event. Further, psychological studies [11] suggest that multimedia data in documents, such as animation, images, or text, contribute to better comprehension for human readers as opposed to a single modality.

Motivated by these observations, we explore usage of image-based features with combination and integration with well-established text classification algorithms to perform multimodal document classification. Multimodal classification is an active research area. Previous work includes video classification

using audio and visual data jointly [12], lyrics-audio synchronization using both text and audio information [13] [14], image classification using figure captions as text in conjunction with image features [15], news video classification using image features along with closed captions as text data [16], etc.

In view of these observations, this thesis explores multimodal document classification via a meta-classification approach. We explore frameworks that integrate state-of-the-art text and image feature-extractors with classification schemes that can assign documents to predetermined categories. Our framework is inspired by Lin and Hauptmann [17], in which unimodal classifiers for each modality are first built, and then second-stage classifiers fuse information from all modalities to make a final decision. We further explore and develop algorithms that fuse information by exploiting correlations not only within a single modality but also across modalities. Techniques that exploit correlations within a modality include image meta-feature vector combination and latent Dirichlet allocation-based image meta-feature extraction. Another technique that exploits correlations between text and image cleans images with text information.

This thesis reports experiments on real-world multimodal documents acquired from Wikipedia and demonstrates the benefits of exploiting multimodalities in classification. In general, classification accuracies are improved by using both text and image features. Further, experiments show that techniques such as image cleaning with text and latent Dirichlet allocation-based image meta-feature extraction are useful in aiding document classification.

This thesis is organized as follows: Chapter 2 introduces essential background regarding document classification and feature extraction. Chapter 3 discusses algorithms for multimodal document classification. Chapter 4 presents our experimental results. Chapter 5 summarizes the main points and the contributions of this thesis.

CHAPTER 2

BACKGROUND

2.1 Supervised Learning and Document Classification

Supervised learning is a topic in machine learning. In supervised learning, there is a teacher and a learner. A teacher provides training data and their corresponding outcomes to a learner, while a learner tries to learn how outcomes are generated from data [18] [19]. Outcomes can be quantitative (such as price of real estate) or categorical (such as it rains/it does not rain.) Whether outcomes are quantitative or categorical depends on the application. Data are usually represented by their features, which are believed to have strong correlations with outcomes. The goal of the learner is to build a prediction model that can predict outcomes with high accuracy given previously unseen data.

For document classification, documents are considered as data and their corresponding categories are considered as outcomes. Categories are categorical outcomes. For example, we can categorize a news document into "finance," "politics," or "sports."

The document classification problem is formally defined as follows. Denote by D the document space that contains all possible documents of interest, and by $C = \{c_1, c_2, \ldots, c_m\}$ a set of m predetermined categories. Each document in D is assigned a real category by an unknown underlying function $g : D \mapsto C$. Given a set of n training documents $d_1, d_2, \ldots, d_n \in D$ and their corresponding categories $l_1, l_2, \ldots, l_n \in C$, design a function $f : D \mapsto C$ such that given a new previously unseen data $d \in D$, the function f predicts the true category $g(d)$ with high accuracy. The function f is called a classifier.

Experiments on a corpus, or a collection of documents, are performed to test the performance of a classification algorithm. Categories of documents in a corpus are usually assigned manually via the following process. For each

document in the corpus, a human will assign a category that is the most relevant to the content of the document. The corpus is then partitioned into two disjoint sets: a training set and a testing set. The training set is processed by learners to generate a classifier, and the testing set is used to evaluate the performance the classifier. Evaluation of goodness of a classifier is addressed in detail in Section 4.1 of this thesis.

In document classification, the original data is too voluminous to be used directly. Feature extraction techniques are used to capture relatively low-dimensional features from intrinsic data. Feature extractors are designed to extract features that are correlated with the outcome category. Feature extraction techniques depend on the application and the nature of the data. The following subsections discuss multimodal documents and the corresponding feature extraction methods.

2.2 Multimodal Documents

Documents are often multimodal. Different modalities originate from different physical sources, yet they usually have correlation with each other. For example, movies consist of video and audio; weather data measurements include temperature, humidity, atomspheric pressure, etc. Modern documents are no exception. While traditional documents are composed of text only, technological advancements have made documents enhanced with image, audio, or video easy to produce; such multimedia documents are more and more popular. Each of these media can be considered as a modality. Since documents are meant to store or communicate information, these modalities are presumably highly correlated with each other. An example is shown in Figure 2.1. In this thesis, "multimodal documents" refers to documents containing text and image.

2.3 Text Feature Extraction

The most popular text document feature extraction method is bag-of-words [10]. To extract features from a corpus, a dictionary is defined as a collection of unique words contained in the corpus of interest. Each unique term in a

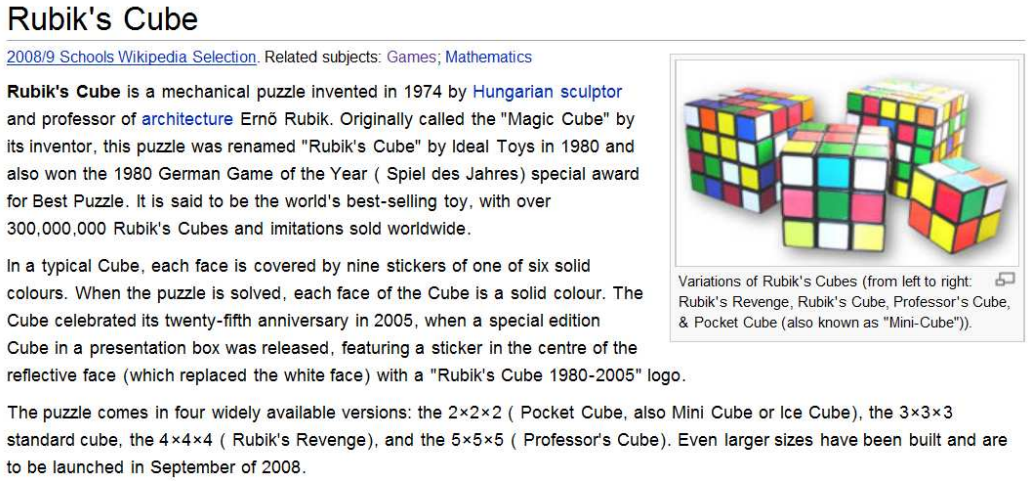

Figure 2.1: An example of multimodal document titled "Rubik's Cube" [20]. The text and the image are highly correlated.

dictionary can be viewed as a feature. The feature vector for a document can then be obtained by counting the occurrence of each term in the dictionary. Given a set of d documents and its corresponding dictionary with t terms, a term-frequency matrix is defined as a $d \times t$ matrix $A \in \mathbf{R}^{d \times t}$. Each element $A(i, j)$ in A, where $i \in \{1, 2, \ldots, d\}$ and $j \in \{1, 2, \ldots, t\}$, is defined as the frequency of the j^{th} word in the dictionary in the i^{th} document of the dataset. The text feature vector for the i^{th} document is the i^{th} row of matrix A.

For example, if we have three documents, $[a\ b\ b\ c]$, $[b\ c\ a\ a]$, and $[b\ b\ c\ d]$, the dictionary is defined as the set $\{a, b, c, d\}$. The term-frequency matrix for this dataset will be:

$$A = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 1 & 0 \\ 0 & 2 & 1 & 1 \end{bmatrix}.$$

However, not all terms in a dictionary are highly correlated with the semantic category of a document. To remove such terms, two main techniques are used. One method is known as stop-words elimination [7], which observes words such as "the" and "and," which convey no discriminative information about the category of a document. Another method is document-frequency thresholding, which removes infrequent words because they are too rare to be used to draw trustworthy interpretations. In our work, these two techniques are applied to a term-frequency matrix to reduce dimensionality of text feature vectors.

Tools for building term-frequency matrices from document collections are widely available [21]. They provide good functionality in removing irrelevant features from a term-frequency matrix. They also provide a variety of post-processing tools for term-frequency matrices, which are useful in representing different types of documents [22].

2.4 Image Feature Extraction

Bag-of-features is a technique which uses local features to represent an image. It has been found to be more useful than global features in image classification and object recognition [23] [24] [25].

The underlying idea can be parallel compared to bag-of-words. The feature for an image is obtained by computing a visual term-frequency vector. However, there are three key differences between visual words and regular words. First, unlike bag-of-words, which uses every word in the document, not every pixel forms a visual word. Only visually significant points of interest are taken into account. Methods for locating points of interest have been actively researched in the computer vision community. Points of interest are typically determined using geometrically meaningful feature point detectors [26] [27]. Remarkably, random sampling techniques have also shown to be useful for determining interest points [28].

Second, for performance reasons, sampled patches are not directly used but are represented by descriptors [29], [30]. Scale-invariant feature transform (SIFT) descriptors [31] are known to be amongst the most competitive image descriptors [30] and are employed in our work. Finally, there is no predetermined dictionary for visual words. Usually, a dictionary is created via training processes using unsupervised learning algorithms, such as hierarchical k-means (HKM) clustering [28] or approximate k-means (AKM) clustering [32]. Each patch is then assigned the visual word that corresponds to its nearest neighbor in the dictionary.

The feature vector generation process is then straightforward. For a set of d images and a visual dictionary with t terms, a visual feature-frequency matrix is defined as a $d \times t$ matrix $B \in \mathbf{R}^{d \times t}$. Each element $B(i, j)$ in B, where $i \in \{1, 2, \ldots, d\}$ and $j \in \{1, 2, \ldots, t\}$, is the frequency of the j^{th} visual word of the dictionary in the i^{th} image of the dataset. The image feature

vector for the i^{th} image is the i^{th} row of B.

CHAPTER 3

CLASSIFICATION ALGORITHMS

As discussed in Chapter 2, text and images both carry significant information that helps the classification process, yet they differ greatly in nature and dimensionality. Also, their correlations are not easy to identify in their native (low-level) form. In this chapter, we explore and discuss classification algorithms that process and exploit both modalities in a unified framework. Our framework is shown in Figure 3.1 and consists of two stages from which we build a meta-classifier.

Meta-classifiers are classifiers that do not directly observe features but observe outputs of base classifiers to make final decisions. Meta-classification approaches first bring multimodal information to a common ground where correlations can be identified and exploited more easily. The first stage processes both modalities individually and synthesizes a meta-feature vector. The second stage observes the meta-features and makes a classification decision. This chapter presents our classification algorithms in this unified framework.

3.1 Concatenate and Classify

The most naive and straightforward method is to concatenate both text and image features as a meta-feature. Due to the characteristics of text and image features, the performance of this kind of classifier is generally dominated by text. In general this method is not robust and depends heavily on how text and image features are generated. The following algorithms and approaches are designed to be more robust to different situations.

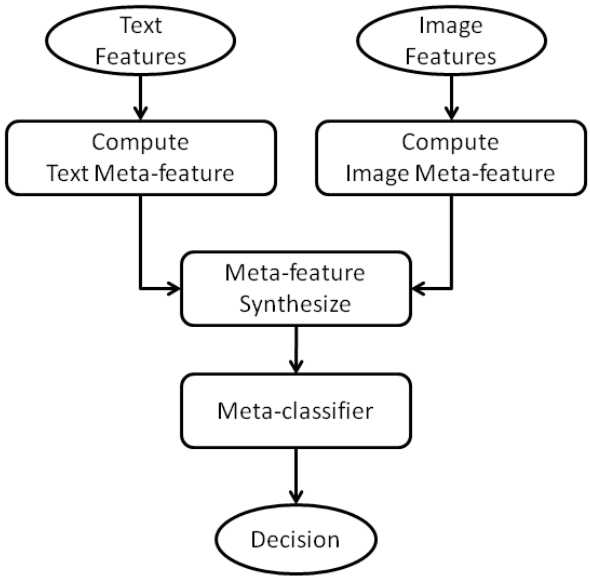

Figure 3.1: Our Unified Framework for Multimodal Document Classification.

3.2 Meta-Classification with Support Vector Machines

Support Vector Machines (SVM)-based meta-classification is illustrated in Figure 3.2. SVM is a widely used learning technique which finds the margin-maximizing hyperplane in a feature space [33]. Margin maximization has strong theoretic backing [34] in structural risk minimization which aims to bound the generalization error of a classifier. The general form of the decision function f of a SVM classifier may be written as

$$f(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i . y_i . K(\mathbf{s}_i, \mathbf{x}) + b, \tag{3.1}$$

where $\mathbf{x}$ is the multi-dimensional feature vector; $\mathbf{s}_i$, $1 \leq i \leq N_s$, the support vectors; N_s, the number of support vectors; and $y_i \in \{1, -1\}$, the true label of the i^{th} training datum. The positive Lagrange multipliers α_i's and hyperplane parameter b are determined by solving the constrained optimization problem that arises from margin maximization. A binary decision of $\{1, -1\}$ is made depending on whether or not $f(\mathbf{x}) > 0$. A soft-output is the exact numerical value of $f(\mathbf{x})$, and its magnitude can be intepreted as the "confidence" of the classifier.

Given a set of training documents, the text and image features are first

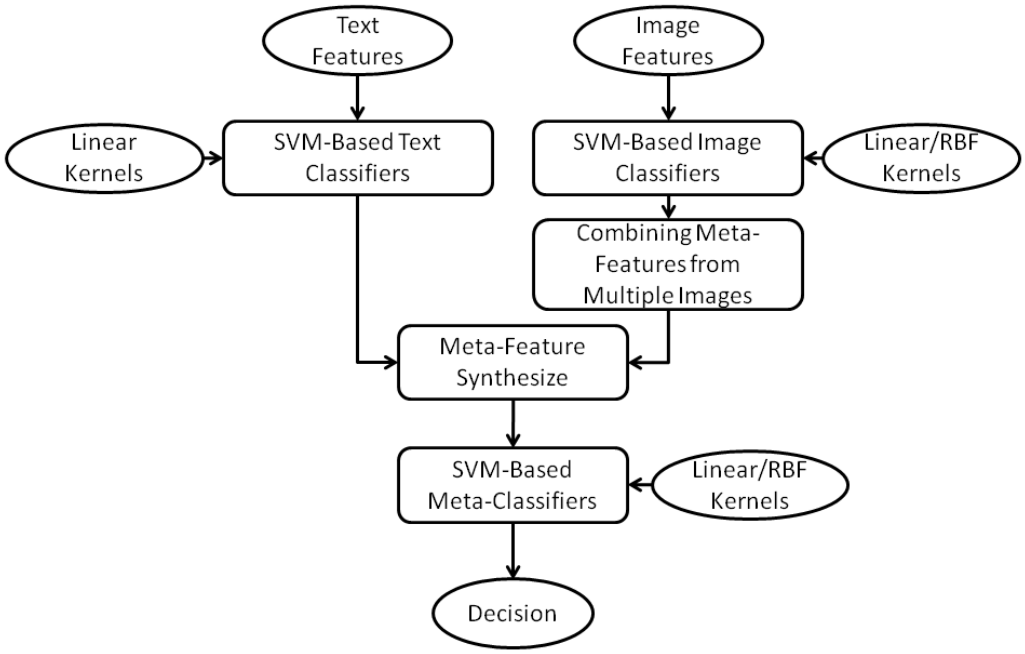

Figure 3.2: SVM-Based Meta-Classification.

extracted using the techniques mentioned in Section 2.3 and Section 2.4 respectively. There are two popular methods to decompose a multiclass classification problem into several binary classification problems. They are referred to as one-against-all and one-against-one [35]. Given a multimodal document database with m categories, to train a multiclass classifier in a one-against-all manner, m base SVM classifiers for text and m base SVM classifiers for images are trained individually. The i^{th} classifier, $i = 1, \ldots, m$, is trained to predict if a testing example (image/text) belongs to the i^{th} category or not. Let a positive soft-output from the i^{th} classifier indicate that a testing example belongs to the i^{th} class. The multiclass classification decision is given as $\hat{f}(\mathbf{x}) = \text{argmax}_{i=1,\ldots,m} f_i(\mathbf{x})$, where $\hat{f}(\mathbf{x})$ is the one-against-all multiclass classifier; f_i, the i^{th} base classifier; $\mathbf{x} \in \mathbf{R}^t$, the feature vector of a testing example; and t, the dimension of a feature vector. In a one-against-one [36] formation, classifiers are trained pairwise between classes. Given new examples, predictions are made based on majority vote from all classifiers. According to Hsu and Lin [37], one-against-one multiclass SVMs are more suitable for practical use than than one-against-all SVMs in solving multiclass classification problems.

To exploit the fact that SVM soft-outputs can convey "confidence" about

the classification results. Instead of using $\hat{f}(\mathbf{x})$, soft-outputs of base classifiers are concatenated and used to represent document features in a new space. This concatenated feature is referred as a meta-feature. Given a multimodal training/testing example and sets of base classifiers of each modality, soft-outputs obtained from base classifiers of each modality are combined into a meta-feature.

For a given document, let $\mathbf{x}_t$ denote the vector of meta-feature from a text-based SVM. An interesting challenge presents itself in constructing the corresponding vector from images, since the number of images in each document may vary. We therefore need a strategy to get a single representative feature vector $\mathbf{x}_i$ for all images in a document.

Given a document with p images, one first obtains meta-feature vectors $\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip}$ by applying image-based SVMs to each of p images. Two strategies are then used to compute $\mathbf{x}_i$. These strategies are summarized in Table 3.1. The first strategy simply computes $\mathbf{x}_i$ as an average of meta-features $\mathbf{x}_i = \sum_{l=0}^{p} \mathbf{x}_{il}$. The implicit assumption is that each image in a document has the same importance.

The above may of course not always be true. Instead, as is observed in practice, a dominant image type may appear in a document. This situation is handled using our second strategy in Table 3.1. We first apply k-means clustering [38] on $\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip}$. Let k be the number of clusters, $\mathbf{x}_{\mu 1}, \ldots, \mathbf{x}_{\mu k}$ be the mean of the clusters, and N_j, $j = 1, 2, .., k$, be the number of images that belong to the k^{th} cluster. The second strategy takes $\mathbf{x}_i = \mathbf{x}_{\mu k'}$ as the representative feature vector, where $\mathbf{x}_{\mu k'}$ is the mean of the largest cluster.

The meta-feature vector for the document $\mathbf{x}_m$ is then defined as the concatenation of $\mathbf{x}_t$ and $\mathbf{x}_i$. Meta-classifiers are then trained on meta-features.

Similar strategies in Table 3.1 can also be applied to visual term-frequency vectors. In such way, the result is a representative image feature vector for a document. Representative image feature vectors are then processed by image-based classifiers to obtain meta-features. We denote the counterpart of strategy 1 on image features as strategy 3 and the counterpart of strategy 2 on image features as strategy 4.

In employing SVMs, the choice of kernel $K(\mathbf{s}_i, \mathbf{x})$ is often important and application/feature dependent. Linear kernels are employed on text features in the first stage (see Figure 3.2), since bag-of-words features are high-dimensional and sparse [10]. Linear kernels and radial basis function (RBF)

kernels both used bag-of-feature features, depending on dimensionality of visual term-frequency vectors. Meta-classifiers are always used with RBF kernels. In general we choose linear kernels for features with higher dimensions and higher sparsity, RBF kernels for features with lower dimensions and lower sparsity [39].

Table 3.1: Strategies for Combining Meta-features from Multiple Images.

Strategy 1: Average
Given: A set of vectors $\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip}$.
Output: $\mathbf{x}_i = \sum_{l=0}^{p} \mathbf{x}_{il}$

Strategy 2: Mean of Largest Cluster
Given: A set of vectors $\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip}$.
Step1: Apply k-means clustering to $\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip}$.
Step2: Find the mean of the largest cluster $\mathbf{x}_{\mu k'}$.
Output: $\mathbf{x}_i = \mathbf{x}_{\mu k'}$

3.3 Meta-Classification via Adaboost

Boosting has its roots in a theoretical framework called probably approximately correct (PAC) learning [40]. The question asked by boosting is whether weak learners that perform slightly better than random guessing can be boosted into a significantly more accurate strong learning algorithm. A breakthrough came in 1995 with the development of the Adaboost algorithm [41], which iteratively determines weights on each weak learner to combine them into a final strong learner.

Adaboost maintains a distribution or set of weights over the training set. Initially, all weights are set equally; but in each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set. Figure 3.3 shows our application of Adaboost to a meta-classifier. Table 3.2 outlines the actual steps used in the Adaboost algorithm. In Table 3.2, S is the training set of meta-features obtained from the first stage of the classifiers in Figure 3.3, and N denotes the total number of training samples. The hypotheses $\{f_t(x)\}_{t=1}^{T}$ correspond to the weak learners which are neural network-based classifiers trained on S, and T denotes the number of weak learners. In our design, typical values of

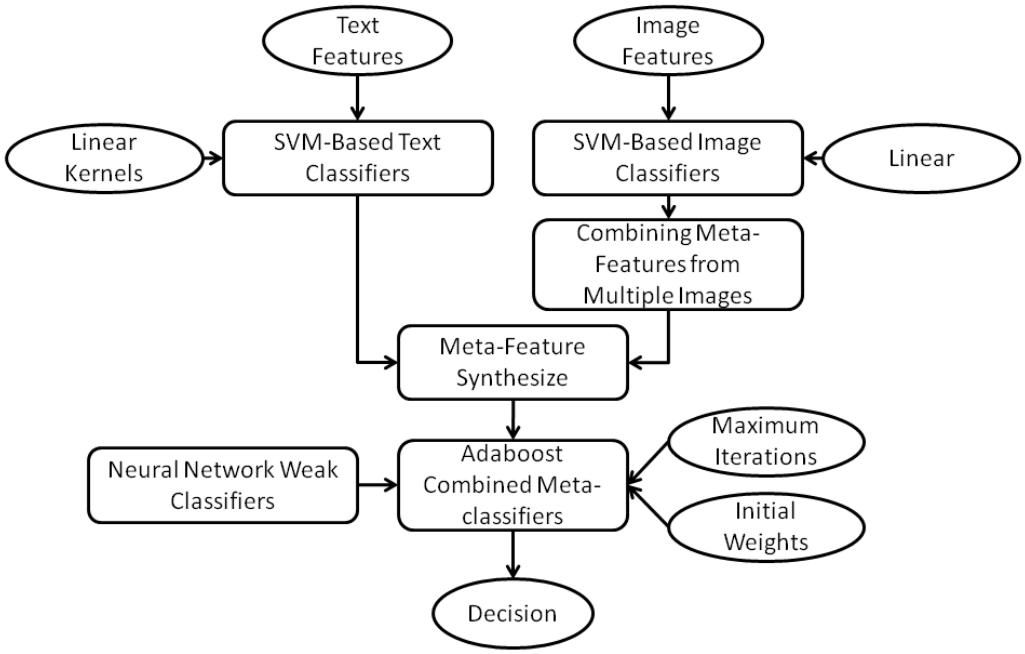

Figure 3.3: Meta-Classification via Adaboost.

T varied from 10 to 20. The final hypothesis $F(x)$ is a weighted majority vote of T weak hypotheses, where β_t is the weight assigned to f_t.

Table 3.2: The Adaboost Learning Algorithm [42].

1. Given (x_i, y_i), $i = 1, 2, \ldots, N$ where $x_i \in S$, $y_i \in \{1, -1\}$
2. Initialize $D_1(i) = \frac{1}{N}$, $i = 1, 2, \ldots, N$
3. Repeat for $t = 1, 2, \ldots, T$:
   (a) Train weak learner using distribution D_t
   (b) Calculate weak hypothesis $f_t : S \longrightarrow \{1, -1\}$ with error ϵ_t
   (c) Choose $\beta_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
   (d) Update $D_t(i)$ to $D_{t+1}(i)$ using β_t, y_i, and $f_t(x_i)$
3. Output decision function $F(x) = sign[\sum_{t=1}^{T} \beta_t f_t(x)]$.

3.4 Cleaning Image with Text Information

As it has been reported in our conference paper [43], image classifiers are always outperformed by text classifiers for the task of document classification. Also, as the image classifiers perform better, the meta-classifiers perform

better as well. This points towards developing techniques that improve image classification performance. In general this is a hard problem, yet in our case we can exploit the fact that we have two correlated modalities.

Data cleaning originated from database applications [44]. Cleaning is used to detect, correct, or remove corrupt or inaccurate data. Data cleaning has been applied to training data cleaning for text classification [45] and computational linguistics [46]. More applications are mentioned in [45]. However, as far as we know, no work has involved in data cleaning for multimodal document classification.

For unimodal documents, a training datum is cleaned by comparing the true label and the prediction. Since true labels are not available for testing data, only training data can be cleaned. For multimodal documents, predictions from one modality are used as the true labels to clean testing data. Text classifiers are generally better and are selected as the baseline to clean images.

Our multimodal image cleaning algorithm is summarized in Table 3.3. Images are picked according to the L_1 distances from image meta-features to the text meta-feature. Only the k images with the smallest L_1 distance are used for further processing. Figure 3.4 illustrates SVM-based meta-classifiers with image cleaning.

Table 3.3: The Image Cleaning Algorithm.

1. Given a text classifier and an image classifier.
   Given a multimodal document with text and p images.
2. Compute the meta-feature for the text $\mathbf{x}_t$ and the images $\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip}$.
3. Compute $D_j = \|\mathbf{x}_t - \mathbf{x}_{ij}\|_1$ for $j = 1, \ldots, p$.
4. Discard all but the k images that correspond to the smallest k D_j's.

To perform image cleaning, we first train both text and image classifiers with all training data. Image cleaning is performed given the aid of the text classifier. A image classifier is then retrained on the cleaned dataset. For testing and training datasets, we have the option of using only cleaned images or the all images. These options are summarized in Table 3.4.

Table 3.4: Meta-Feature Computation Options in Our Image Cleaning Framework.

	Use Cleaned Training Data?	Use Cleaned Testing Data?
1	Yes	No
2	Yes	No
3	No	Yes
4	No	No

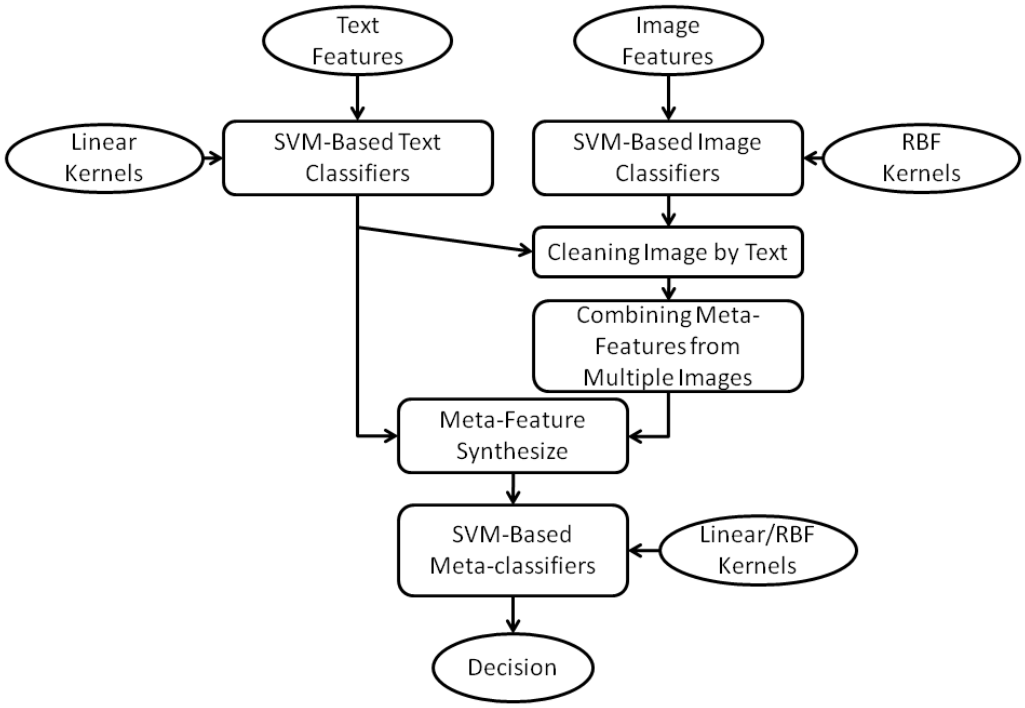

Figure 3.4: SVM-Based Meta-Classification with Image Cleaning.

3.5 Latent Dirichlet Allocation-Based Image Meta-Feature Extraction

Latent Dirichlet allocation (LDA) is a generative topic model that employs a set of hidden topic variables to explain observed term-frequency data [47]. LDA shares similar ideas with other topic modeling techniques such as latent semantic indexing (LSI) [48] and probabilistic latent semantic indexing (pLSI) [49]. In LDA, each document is viewed as a mixture of various topics, where topic distributions are assumed to follow a Dirichlet prior. Figure 3.5 shows the graphical model of LDA. There $\mathbf{w}$ is the term-frequency vector of a document, $\mathbf{z}$ are hidden topic variables, θ are the parameters of the top-

ics, β are the probabilities of generating a word given a topic, and α is the parameter for a Dirichlet distribution.

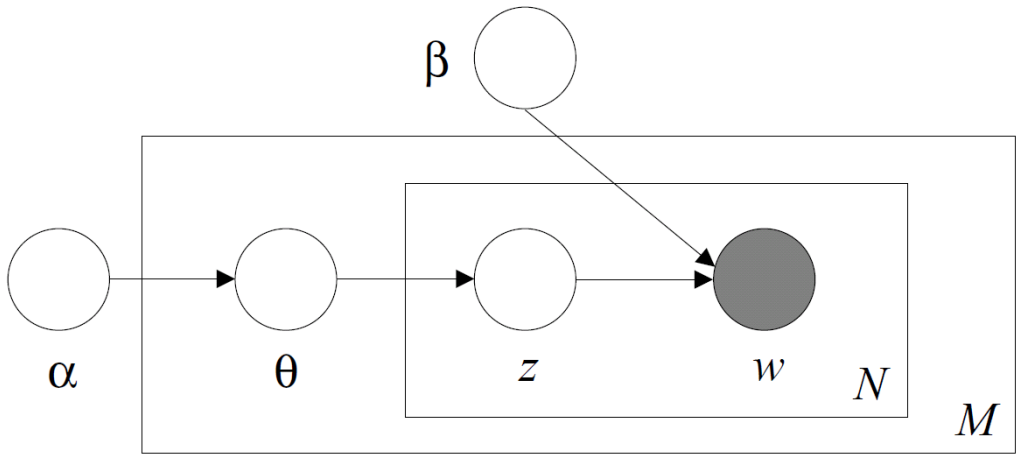

Figure 3.5: The graphical model of the LDA.

For example, consider an LDA model with two topics: "CAT" and "DOG." Topic "CAT" has higher probability of generating words such as "milk," "meow," "kitten," and "cat." Topic "DOG" likewise has higher probability of generating words such as "puppy," "bark," and "bone." Words without special relevance to these two topics, such as "computer," "dividend," "and," or "that," will have roughly even probability under "CAT" and "DOG." LDA assumes that each document is generated by a four-step process: (1) pick the number of words from a Poisson distribution, (2) pick a multinomial distribution over topics from a Dirichlet distribution, (3) pick a topic for each word, and (4) pick each word individually from a multinomial probability distribution conditioned on the topic.

LDA is widely used in the area of text mining [50] [51]. Application of topic modeling techniques to a visual term-frequency matrix is also popular. Fei-fei [52] used a variant of LDA to learn natural scene categories. Sivic et al. [53] used pLSI and LDA models to discover object categories in a set of unlabeled images.

Bag-of-features feature representation originated from image classification, where each image contains a certain type of object. This is, however, not the case in multimodal documents. Multimodal documents often contain more than one type of images. For example, an article about a city may contain photos of the city, maps of the city, the flag of the city, or even the

photo of the current mayor of the city. Figure 3.6 shows an example of a multimodal document about a city. In view of the characteristics of images in multimodal documents, LDA is applied to visual term-frequency matrices. Analogously to [52] [53], we can view each type of image as a topic and find the distribution of the topics (types of image) for each document. The image topic distribution is then taken as image features. Image features are further processed by SVMs to generate meta-features that share the same common ground as the text-based meta-features. Image meta-features are then concatenated with text meta-features to perform meta-classification. The integration between LDA based meta-features and our framework is depicted in Figure 3.7.

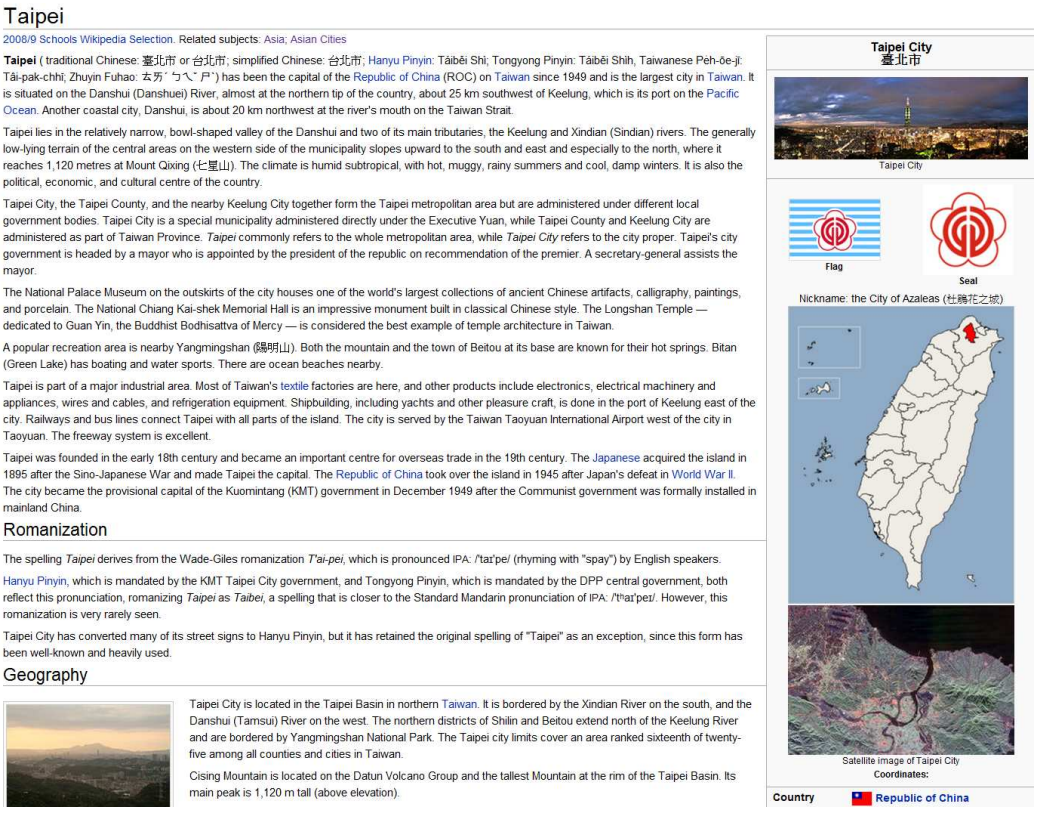

Figure 3.6: An example of multimodal document titled "Taipei" [54]. This document contains a satellite image, a map, a flag, a seal, and the city's photos.

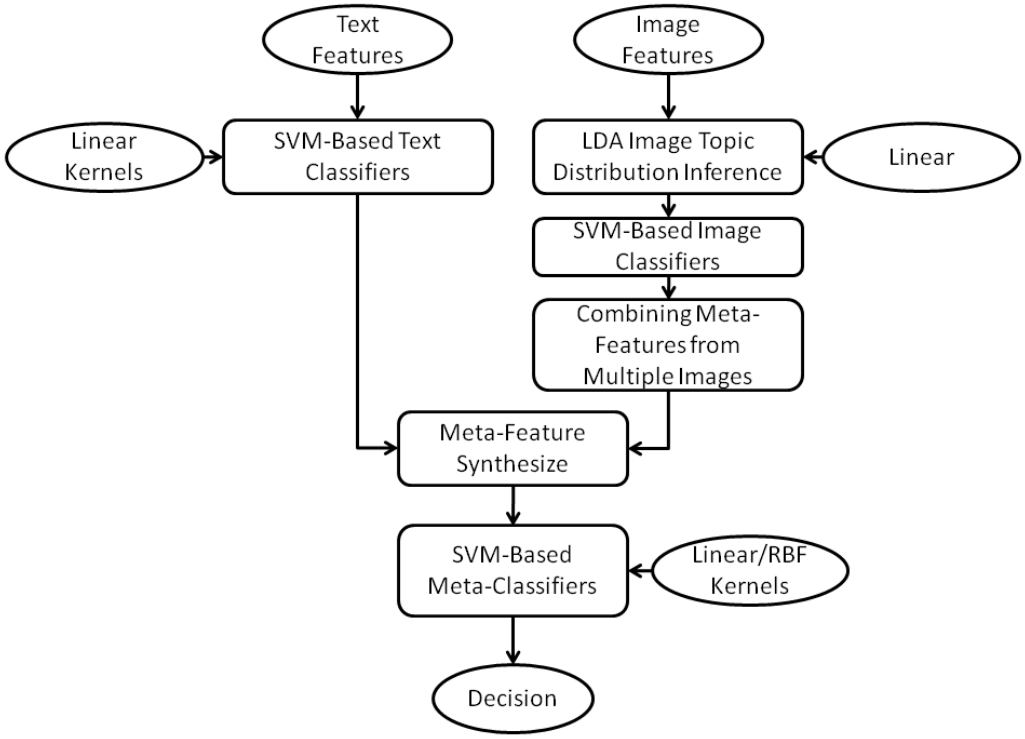

Figure 3.7: LDA Image Topic Meta-Feature extraction with SVM-Based Meta-Classification.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Performance Evaluation

Recall, precision, and F_1-score are widely used to evaluate the performance of classification algorithms [10].

Let $f(\mathbf{x})$ denote the designed classifier, m the number of classes, and $\mathbf{x}_j$ the j^{th} test sample, whose true label is $y_j \in \{1, 2, \ldots, m\}$. We define:

- f_{++}: the number of testing data for which $f(\mathbf{x}_j) = k$ and $y_j = k$,
- f_{+-}: the number of testing data for which $f(\mathbf{x}_j) = k$ and $y_j \neq k$, and
- f_{-+}: the number of testing data for which $f(\mathbf{x}_j) \neq k$ and $y_j = k$.

Recall and precision for the k^{th} class, where $k \in \{1, 2, \ldots, m\}$, are defined as:

$$Recall_k(f) = \frac{f_{++}}{f_{++} + f_{-+}}$$

$$Precision_k(f) = \frac{f_{++}}{f_{++} + f_{+-}}.$$

Clearly, there is a trade-off between the two quantities. The F_1-score, is the harmonic mean of recall and precision and is often used to evaluate the performance of the classifier:

$$F_{1k}(f) = \frac{2Recall_k(f)Precision_k(f)}{Recall_k(f) + Precision_k(f)}.$$

The higher F_{1k} is, the better the classification performance is. We use F_1-scores to evaluate the performance of our algorithms.

4.2 Dataset Collection

A key challenge in evaluating multimodal document classification algorithms is the choice of a suitable real-world dataset. Unlike conventional text-based document classification where there are well-established benchmarks such as 20 Newsgroups [55] and Reuters-21578 [56], real-world benchmarks are not readily available for multimodal document classification.

Results on synthetic datasets are not convincing and can often be misleading. To work with real-world multimodal documents, we have used web-page documents from Wikipedia selection for schools 2008/2009 edition [57]. This database contains around 5,500 articles and is about the size of a 20-volume encyclopedia. Each document contains text and a number of images generally ranging from 3 to 20. As we conjectured earlier, the images can be qualitatively seen to relate well to the text in a document. For example, an article titled "Clarinet" [58] from "music" category contains images of clarinets and people playing clarinets. Also, different categories contain different types of images. As shown in Figure 4.1, documents in "musical instruments" may contain images of musical instruments and notes, while documents in "performers and composers" may contain images of people, as shown in Figure 4.2. The statistics and details of the database are listed in Table 4.1.

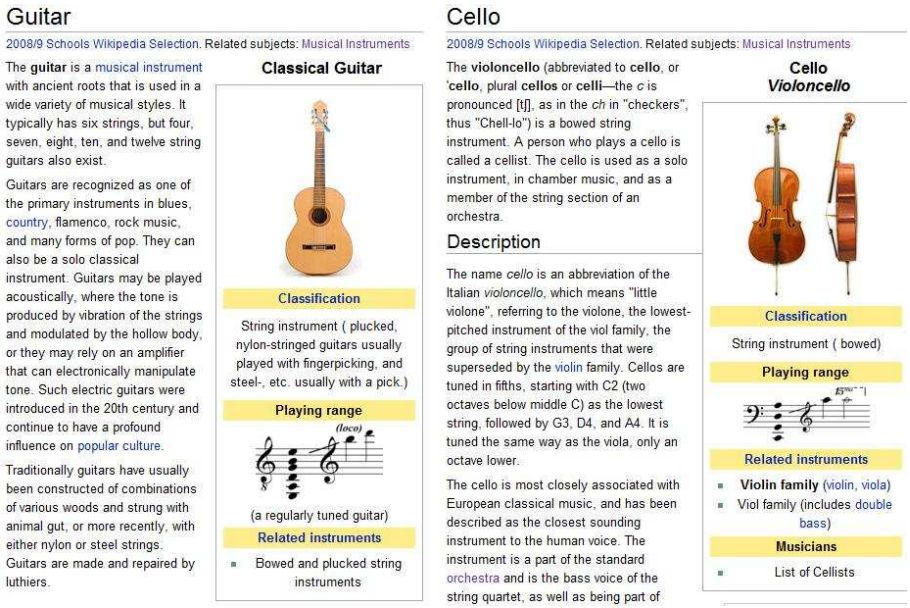

Figure 4.1: Sample documents from "musical instruments."

The advantages of working with the Wikipedia database are that it is

Figure 4.2: Sample documents from "performers and composers."

widely used, easily accessible, and comes with manually preassigned labels. The significant challenge, however, is that many document categories contain a small number of documents, and that means fewer documents are available for training.

4.3 Results

We experimented with the strategies discussed in Section 3.2 and 3.3 with the first four data collections listed in Table 4.2. These results are reported in our conference paper [43]. The first two sets (labeled 1 and 2) are composed of three to four categories. In realistic databases, sometimes categories can be a subset of main categories. Two following document collections (labeled 3 and 4) are created with subsets of an umbrella category. The fifth and sixth datasets are explained later.

We randomly partition document collections into training and testing sets. Due to the limited number of documents, ten documents are used as training data for each category. Multiclass classification is done in a one-against-all manner.

Recall, precision, and their harmonic mean F_1 are reported for each class in Table 4.3 for document collection 1. Analogous results for document collec-

Table 4.1: Categories of Wikipedia Selection for Schools.

Name	Number of Documents	Number of Images	Number of Subcategories
art	86	602	1
business	141	1273	4
citizenship	311	2415	9
design and technology	282	2655	5
everyday life	430	3882	11
geography	1198	19273	17
history	836	6388	11
information technology	86	354	5
language and literature	197	842	8
math	263	1027	1
music	168	654	4
people	751	5326	19
religion	176	1029	6
science	1205	7226	3

tions 2 to 4 are reported in Tables 4.4 – 4.6. In these tables, R_I, P_I represent recall and precision for document classification based on image features only. Likewise, R_T, P_T represent recall and precision of document classification based on text features only. The F_1-scores F_I, F_T are analogous. Symbols with subscripts $ADA-i, i = 1, 2$ represent meta-classification results using Adaboost with image meta-feature vector combination strategy i (as defined in Table 3.1), and symbols with subscripts $SVM-i$ represent analogous results using the SVM-based meta-classifier. MAX_i stands for maximum improvement for the corresponding image feature vector combination strategy i. This quantity is given by

$$\max(F_{ADA-i} - \max(F_T, F_I), F_{SVM-i} - \max(F_T, F_I)).$$

The numbers after $\pm$ in Tables 4.3 – 4.6 are estimated standard deviations and are given by

$$\sigma_i = \sqrt{\frac{p_i(1-p_i)}{n_i - 1}},$$

where p_i is the recall, precision, or F_1-score for class i, and n_i is the number of documents in class i.

Improvements are observed from both SVM- and Adaboost-based meta-

Table 4.2: Selected Datasets for Wikipedia.

Set		Name	Number of Documents	Number of Images
1	1.1	art	86	602
	1.2	business	141	1273
	1.3	information technology	86	354
	1.4	religion	176	1029
2	2.1	design and technology	282	2655
	2.2	math	263	1027
	2.3	music	168	654
3	3.1	science.biology	771	4526
	3.2	science.chemistry	162	793
	3.3	science.physics	251	1907
4	4.1	music.musical genres styles eras and events	31	146
	4.2	music.musical instruments	37	220
	4.3	music.musical recordings and compositions	33	37
	4.4	music.performers and composers	67	251
5	5.1	geography	19273	602
	5.2	history	836	6388
	5.3	people	751	5326
	5.4	science	1205	7226
6	6.1	geography	19273	602
	6.2	science	1205	7226

Table 4.3: Recall, Precision, and F_1-score for Set 1 [43].

	1.1	1.2	1.3	1.4	Average
R_I	$51.7 \pm 5.8\%$	$25 \pm 3.8\%$	$37.4 \pm 5.6\%$	$61.4 \pm 3.8\%$	43.9%
R_T	$84.5 \pm 4.2\%$	$98.6 \pm 1.0\%$	$68.2 \pm 5.4\%$	$69.3 \pm 3.6\%$	80.2%
R_{ADA-1}	$96.6 \pm 2.1\%$	$86.1 \pm 3.0\%$	$76.6 \pm 4.9\%$	$85 \pm 2.8\%$	86.1%
R_{ADA-2}	$87.9 \pm 3.8\%$	$97.2 \pm 1.4\%$	$68.2 \pm 5.4\%$	$78 \pm 3.2\%$	82.8%
R_{SVM-1}	$91.4 \pm 3.2\%$	$97.2 \pm 1.4\%$	$66.4 \pm 5.5\%$	$76.4 \pm 3.3\%$	82.8%
R_{SVM-2}	$87.9 \pm 3.8\%$	$97.2 \pm 1.4\%$	$62.6 \pm 5.6\%$	$80.3 \pm 3.1\%$	82.0%
P_I	$41.1 \pm 5.7\%$	$32.1 \pm 4.1\%$	$50 \pm 5.8\%$	$50.3 \pm 3.9\%$	43.4%
P_T	$90.7 \pm 3.4\%$	$48.3 \pm 4.4\%$	$98.7 \pm 1.3\%$	$98.9 \pm 0.8\%$	84.1%
P_{ADA-1}	$86.2 \pm 4.0\%$	$67.4 \pm 4.1\%$	$96.5 \pm 2.1\%$	$88.5 \pm 2.5\%$	84.6%
P_{ADA-2}	$82.3 \pm 4.4\%$	$63.6 \pm 4.2\%$	$91.3 \pm 3.3\%$	$88.4 \pm 2.5\%$	81.4%
P_{SVM-1}	$80.3 \pm 4.6\%$	$62.5 \pm 4.2\%$	$92.2 \pm 3.1\%$	$89 \pm 2.4\%$	81.0%
P_{SVM-2}	$77.3 \pm 4.8\%$	$63.1 \pm 4.2\%$	$93.1 \pm 2.9\%$	$88.7 \pm 2.5\%$	80.5%
F_I	$45.8 \pm 5.8\%$	$28.1 \pm 3.9\%$	$42.8 \pm 5.7\%$	$55.3 \pm 3.9\%$	43.0%
F_T	$87.5 \pm 3.8\%$	$64.8 \pm 4.2\%$	$80.7 \pm 4.6\%$	$81.5 \pm 3.0\%$	78.6%
F_{ADA-1}	$91.1 \pm 3.3\%$	$75.6 \pm 3.8\%$	$85.4 \pm 4.1\%$	$86.7 \pm 2.6\%$	84.7%
F_{ADA-2}	$85 \pm 4.1\%$	$76.9 \pm 3.7\%$	$78.1 \pm 4.8\%$	$82.8 \pm 2.9\%$	80.7%
F_{SVM-1}	$85.5 \pm 4.1\%$	$76.1 \pm 3.7\%$	$77.2 \pm 4.8\%$	$82.2 \pm 3.0\%$	80.2%
F_{SVM-2}	$82.3 \pm 4.4\%$	$76.5 \pm 3.7\%$	$74.9 \pm 5.0\%$	$84.3 \pm 2.8\%$	79.5%
MAX_1	3.6%	11.2%	4.8%	5.3%	6.2%
MAX_2	-2.5%	12.1%	-1.8%	1.4%	2.3%

classifiers, validating our intuition that augmenting classifier decisions with image features can improve document classification results and does not acutely depend on the choice of the classifier. We can also observe that the best strategy for combining features from multiple images depends on the document collection. Given a new document collection, prior knowledge of the collection may be useful for picking a strategy beforehand.

If we view recall, precision, or F_1-score as estimates of the unknown parameter of a Bernoulli distribution, the formula

$$\sigma_i = \sqrt{\frac{p_i(1-p_i)}{n_i - 1}}$$

gives the estimate of the standard deviation of recall, precision, or F_1-score. It is more desirable to have higher improvement-to-standard-deviation ratio.

The previous analysis suggests using larger datasets. The fifth and sixth datasets are therefore compiled by categories that contain more than 500

Table 4.4: Recall, Precision, and F_1-score for Set 2 [43].

	2.1	2.2	2.3	Average
R_I	$91.7 \pm 1.7\%$	$75.5 \pm 2.7\%$	$43.9 \pm 3.1\%$	70.4%
R_T	$65.6 \pm 2.9\%$	$94.3 \pm 1.5\%$	$69.9 \pm 2.9\%$	76.6%
R_{ADA-1}	$93.7 \pm 1.5\%$	$90.6 \pm 1.8\%$	$78.9 \pm 2.5\%$	87.7%
R_{ADA-2}	$91.3 \pm 1.7\%$	$91.7 \pm 1.7\%$	$73.2 \pm 2.8\%$	85.4%
R_{SVM-1}	$92.1 \pm 1.6\%$	$91.2 \pm 1.8\%$	$82.9 \pm 2.3\%$	88.7%
R_{SVM-2}	$92.5 \pm 1.6\%$	$90.6 \pm 1.8\%$	$78.1 \pm 2.6\%$	87.1%
P_I	$71.6 \pm 2.7\%$	$94.2 \pm 1.5\%$	$60 \pm 3.1\%$	75.3%
P_T	$95.4 \pm 1.3\%$	$61.4 \pm 3.1\%$	$86.9 \pm 2.1\%$	81.2%
P_{ADA-1}	$88.1 \pm 2.0\%$	$96.7 \pm 1.1\%$	$81.5 \pm 2.4\%$	88.8%
P_{ADA-2}	$86.8 \pm 2.1\%$	$89.3 \pm 1.9\%$	$85.7 \pm 2.2\%$	87.3%
P_{SVM-1}	$90.3 \pm 1.8\%$	$94.6 \pm 1.4\%$	$81.6 \pm 2.4\%$	88.8%
P_{SVM-2}	$87.6 \pm 2.0\%$	$91.1 \pm 1.8\%$	$87.3 \pm 2.1\%$	88.7%
F_I	$80.4 \pm 2.4\%$	$83.8 \pm 2.3\%$	$50.7 \pm 3.1\%$	71.6%
F_T	$77.7 \pm 2.5\%$	$74.3 \pm 2.8\%$	$77.5 \pm 2.6\%$	76.5%
F_{ADA-1}	$90.8 \pm 1.8\%$	$93.6 \pm 1.5\%$	$80.2 \pm 2.5\%$	88.2%
F_{ADA-2}	$89 \pm 1.9\%$	$90.5 \pm 1.8\%$	$78.9 \pm 2.5\%$	86.1%
F_{SVM-1}	$91.2 \pm 1.7\%$	$92.8 \pm 1.6\%$	$82.3 \pm 2.4\%$	88.8%
F_{SVM-2}	$90 \pm 1.8\%$	$90.9 \pm 1.8\%$	$82.4 \pm 2.4\%$	87.8%
MAX_1	10.8%	9.7%	4.8%	8.4%
MAX_2	8.6%	6.7%	3.5%	6.3%

documents. Statistics of the datasets are summarized in Table 4.2. The fifth dataset contains 4 categories, and the sixth dataset contains two categories.

Experiments with 50 training documents per category are performed. Each category is randomly partitioned into a training dataset and a testing dataset. 10 and 20 runs are performed on dataset 5 and 6, respectively. Multiclass classification is done in a one-against-one manner. To evaluate the performance of image cleaning techniques and LDA-based meta-features, we take image meta-feature vector combination strategy 1 as the baseline. Image cleaning and LDA-based image meta-feature extraction are experimented with using the baseline strategy.

Average results are summarized in Table 4.7 and Table 4.8. We also summarize improvements of F_1-score in Table 4.9. We define ΔF as a shorthand for $F - F_T$. The numbers after $\pm$ are the sample standard deviations.

Following conventions described earlier, R, P, F represent recall, precision, and F_1-score, respectively. Subscript $CON - i$ represents concatenating fea-

Table 4.5: Recall, Precision, and F_1-score for Set 3 [43].

	3.1	3.2	3.3	Average
R_I	$17.2 \pm 1.4\%$	$58.9 \pm 4.0\%$	$82 \pm 2.5\%$	52.7%
R_T	$92.9 \pm 0.9\%$	$61 \pm 4.0\%$	$55.3 \pm 3.2\%$	69.7%
R_{ADA-1}	$86.3 \pm 1.2\%$	$92.9 \pm 2.1\%$	$73.7 \pm 2.8\%$	84.3%
R_{ADA-2}	$86 \pm 1.3\%$	$90.8 \pm 2.4\%$	$72.4 \pm 2.9\%$	83.0%
R_{SVM-1}	$88.9 \pm 1.1\%$	$78.7 \pm 3.3\%$	$76 \pm 2.8\%$	81.2%
R_{SVM-2}	$88.5 \pm 1.2\%$	$76.6 \pm 3.4\%$	$72.8 \pm 2.9\%$	79.3%
P_I	$88.6 \pm 1.2\%$	$57.2 \pm 4.0\%$	$22.4 \pm 2.7\%$	56.1%
P_T	$83.8 \pm 1.3\%$	$65.2 \pm 3.9\%$	$81.6 \pm 2.5\%$	76.9%
P_{ADA-1}	$94.3 \pm 0.8\%$	$60.6 \pm 4.0\%$	$78.8 \pm 2.6\%$	77.9%
P_{ADA-2}	$93.4 \pm 0.9\%$	$59.5 \pm 4.0\%$	$78.5 \pm 2.7\%$	77.1%
P_{SVM-1}	$92.4 \pm 1.0\%$	$70.3 \pm 3.7\%$	$72.7 \pm 2.9\%$	78.4%
P_{SVM-2}	$91.8 \pm 1.0\%$	$67.5 \pm 3.8\%$	$70.5 \pm 2.9\%$	76.6%
F_I	$28.8 \pm 1.6\%$	$58 \pm 4.0\%$	$35.2 \pm 3.1\%$	40.7%
F_T	$88.1 \pm 1.2\%$	$63 \pm 3.9\%$	$65.9 \pm 3.1\%$	72.4%
F_{ADA-1}	$90.1 \pm 1.1\%$	$73.4 \pm 3.6\%$	$76.2 \pm 2.7\%$	79.9%
F_{ADA-2}	$89.5 \pm 1.1\%$	$71.9 \pm 3.7\%$	$75.3 \pm 2.8\%$	78.9%
F_{SVM-1}	$90.6 \pm 1.1\%$	$74.2 \pm 3.6\%$	$74.3 \pm 2.8\%$	79.7%
F_{SVM-2}	$90.1 \pm 1.1\%$	$71.8 \pm 3.7\%$	$71.7 \pm 2.9\%$	77.8%
MAX_1	2.5%	11.2%	10.3%	8.0%
MAX_2	1.4%	8.9%	9.4%	6.6%

tures directly from each modality (Section 3.1) using image meta-feature vector combination strategy i. Subscript $SVM-i$ represents SVM-based meta-classifiers (Section 3.2) using image meta-feature vector combination strategy i. $CLN-j$ represents image cleaning techniques (Section 3.4) with the j^{th} cleaning option listed in Table 3.4. LDA refers to using LDA-based image meta-feature-extractors (Section 3.5).

We make the following observations. Concatenating feature vectors directly is not robust. The performance is dominated by text, since text features are favored by the linear kernel, which is used in our experiments. Concatenating feature vectors does not show improvements neither in dataset 5 nor in dataset 6.

Improvements are seen in row 3 to row 11 in Table 4.9. Meta-classifiers successfully exploited the fact that text and image features consist of correlated or complementary information; hence, combining both modalities increases performance. Comparing among ΔF_{SVM-1} through ΔF_{SVM-4}, we conclude

Table 4.6: Recall, Precision, and F_1-score for Set 4 [43].

	4.1	4.2	4.3	4.4	Average
R_I	$15.4 \pm 8.1\%$	$100 \pm 0.0\%$	$40 \pm 10.4\%$	$63.3 \pm 6.4\%$	54.7%
R_T	$69.2 \pm 10.3\%$	$92.6 \pm 5.1\%$	$80 \pm 8.5\%$	$67.3 \pm 6.3\%$	77.3%
R_{ADA-1}	$53.8 \pm 11.1\%$	$100 \pm 0.0\%$	$80 \pm 8.5\%$	$79.6 \pm 5.4\%$	78.4%
R_{ADA-2}	$53.9 \pm 11.1\%$	$100 \pm 0.0\%$	$100 \pm 0.0\%$	$77.6 \pm 5.6\%$	82.9%
R_{SVM-1}	$69.2 \pm 10.3\%$	$96.3 \pm 3.7\%$	$40 \pm 10.4\%$	$87.8 \pm 4.4\%$	73.3%
R_{SVM-2}	$69.2 \pm 10.3\%$	$96.3 \pm 3.7\%$	$80 \pm 8.5\%$	$83.7 \pm 4.9\%$	82.3%
P_I	$50 \pm 11.2\%$	$55.1 \pm 9.8\%$	$33.3 \pm 10.0\%$	$88.6 \pm 4.2\%$	56.8%
P_T	$69.2 \pm 10.3\%$	$71.5 \pm 8.9\%$	$30.8 \pm 9.8\%$	$100 \pm 0.0\%$	67.9%
P_{ADA-1}	$87.5 \pm 7.4\%$	$75 \pm 8.5\%$	$40 \pm 10.4\%$	$97.5 \pm 2.1\%$	75.0%
P_{ADA-2}	$70 \pm 10.2\%$	$73 \pm 8.7\%$	$71.4 \pm 9.6\%$	$95 \pm 2.9\%$	77.4%
P_{SVM-1}	$75 \pm 9.7\%$	$89.7 \pm 6.0\%$	$50 \pm 10.7\%$	$87.8 \pm 4.4\%$	75.6%
P_{SVM-2}	$75 \pm 9.7\%$	$81.3 \pm 7.6\%$	$66.7 \pm 10.0\%$	$93.2 \pm 3.4\%$	79.0%
F_I	$23.5 \pm 9.5\%$	$71.1 \pm 8.9\%$	$36.4 \pm 10.3\%$	$73.8 \pm 5.9\%$	51.2%
F_T	$69.2 \pm 10.3\%$	$80.7 \pm 7.7\%$	$44.4 \pm 10.6\%$	$80.5 \pm 5.3\%$	68.7%
F_{ADA-1}	$66.7 \pm 10.5\%$	$85.7 \pm 6.9\%$	$53.3 \pm 10.6\%$	$87.6 \pm 4.4\%$	73.3%
F_{ADA-2}	$60.9 \pm 10.9\%$	$84.4 \pm 7.1\%$	$83.3 \pm 8.0\%$	$85.4 \pm 4.7\%$	78.5%
F_{SVM-1}	$72 \pm 10.0\%$	$92.9 \pm 5.0\%$	$44.4 \pm 10.6\%$	$87.8 \pm 4.4\%$	74.3%
F_{SVM-2}	$72 \pm 10.0\%$	$88.1 \pm 6.4\%$	$72.7 \pm 9.5\%$	$88.2 \pm 4.3\%$	80.3%
MAX_1	2.8%	12.2%	8.9%	7.3%	7.8%
MAX_2	2.8%	3.7%	38.9%	8.6%	13.5%

that the best strategy depends on the document collection. There is no best strategy for image meta-feature vector combination. Comparing ΔF_{SVM-1} with ΔF_{CLN-1}, ΔF_{CLN-2}, ΔF_{CLN-3}, ΔF_{CLN-4} and ΔF_{LDA}, improvements are observed from dataset 5 while performance degrades in dataset 6. This difference is due to the nature of datasets. Datasets with different natures benefit from different strategies and techniques.

Table 4.7: Recall, Precision, and F_1-score for Set 5.

	5.1	5.2	5.3	5.4	Average
R_I	23.7%	26.4%	35.9%	37.7%	30.9%
R_T	85.1%	70.3%	74.1%	90.4%	80.0%
R_{CON-1}	84.5%	69.9%	75.0%	89.7%	79.8%
R_{CON-2}	84.6%	69.9%	74.1%	89.6%	79.6%
R_{SVM-1}	89.5%	71.3%	71.8%	87.0%	79.9%
R_{SVM-2}	90.0%	68.7%	72.2%	87.5%	79.6%
R_{SVM-3}	84.8%	73.0%	75.1%	91.2%	81.0%
R_{SVM-4}	84.8%	73.1%	75.1%	91.1%	81.0%
R_{CLN-1}	85.5%	70.5%	77.9%	91.9%	81.4%
R_{CLN-2}	85.8%	70.6%	77.8%	91.3%	81.4%
R_{CLN-3}	86.4%	71.2%	75.3%	91.0%	81.0%
R_{CLN-4}	86.6%	70.6%	75.6%	91.0%	80.9%
R_{LDA}	84.1%	73.0%	77.0%	91.7%	81.5%
P_I	63.5%	20.6%	24.9%	31.3%	35.1%
P_T	91.6%	69.4%	75.4%	81.2%	79.4%
P_{CON-1}	92.1%	69.4%	73.9%	80.3%	78.9%
P_{CON-2}	92.1%	68.8%	74.1%	80.0%	78.7%
P_{SVM-1}	88.8%	70.3%	76.6%	87.5%	80.8%
P_{SVM-2}	87.7%	72.3%	76.2%	86.8%	80.7%
P_{SVM-3}	93.3%	69.3%	75.4%	82.1%	80.0%
P_{SVM-4}	93.3%	69.3%	75.5%	82.1%	80.0%
P_{CLN-1}	93.4%	71.0%	74.8%	82.4%	80.4%
P_{CLN-2}	93.1%	71.3%	74.7%	82.7%	80.4%
P_{CLN-3}	91.9%	71.4%	76.6%	82.4%	80.6%
P_{CLN-4}	91.9%	72.1%	76.3%	82.2%	80.6%
P_{LDA}	93.6%	69.5%	75.6%	81.8%	80.1%
F_I	31.1%	19.4%	25.1%	27.9%	25.9%
F_T	88.1%	69.8%	74.6%	85.4%	79.5%
F_{CON-1}	88.1%	69.5%	74.3%	84.6%	79.1%
F_{CON-2}	88.1%	69.3%	73.9%	84.4%	78.9%
F_{SVM-1}	89.0%	70.3%	73.9%	87.1%	80.1%
F_{SVM-2}	88.8%	70.2%	73.9%	87.0%	80.0%
F_{SVM-3}	88.8%	71.1%	75.0%	86.2%	80.3%
F_{SVM-4}	88.8%	71.1%	75.1%	86.2%	80.3%
F_{CLN-1}	89.2%	70.6%	76.1%	86.8%	80.7%
F_{CLN-2}	89.2%	70.8%	76.0%	86.7%	80.7%
F_{CLN-3}	89.0%	71.1%	75.7%	86.3%	80.5%
F_{CLN-4}	89.1%	71.1%	75.8%	86.2%	80.6%
F_{LDA}	88.6%	71.1%	76.2%	86.4%	80.6%

Table 4.8: Recall, Precision, and F_1-score for Set 6.

	6.1	6.2	Average
R_I	33.2%	78.9%	56.1%
R_T	89.1%	95.8%	92.4%
R_{CON-1}	88.7%	94.6%	91.7%
R_{CON-2}	87.9%	94.8%	91.4%
R_{SVM-1}	93.3%	93.0%	93.1%
R_{SVM-2}	93.3%	92.5%	92.9%
R_{SVM-3}	93.2%	92.3%	92.7%
R_{SVM-4}	92.8%	92.7%	92.8%
R_{CLN-1}	93.1%	92.6%	92.9%
R_{CLN-2}	93.2%	92.8%	93.0%
R_{CLN-3}	93.0%	92.6%	92.8%
R_{CLN-4}	93.3%	92.2%	92.8%
R_{LDA}	91.6%	93.8%	92.7%
P_I	77.5%	40.7%	59.1%
P_T	97.4%	83.7%	90.6%
P_{CON-1}	96.7%	83.0%	89.8%
P_{CON-2}	96.8%	82.1%	89.4%
P_{SVM-1}	95.9%	88.9%	92.4%
P_{SVM-2}	95.6%	88.9%	92.3%
P_{SVM-3}	95.5%	88.7%	92.1%
P_{SVM-4}	95.7%	88.3%	92.0%
P_{CLN-1}	95.7%	88.7%	92.2%
P_{CLN-2}	95.8%	88.8%	92.3%
P_{CLN-3}	95.7%	88.5%	92.1%
P_{CLN-4}	95.5%	88.9%	92.2%
P_{LDA}	96.3%	86.7%	91.5%
F_I	42.4%	51.5%	46.9%
F_T	93.0%	89.3%	91.1%
F_{CON-1}	92.5%	88.4%	90.4%
F_{CON-2}	92.1%	87.9%	90.0%
F_{SVM-1}	94.5%	90.8%	92.7%
F_{SVM-2}	94.4%	90.6%	92.5%
F_{SVM-3}	94.3%	90.4%	92.3%
F_{SVM-4}	94.2%	90.3%	92.3%
F_{CLN-1}	94.4%	90.5%	92.4%
F_{CLN-2}	94.4%	90.6%	92.5%
F_{CLN-3}	94.3%	90.4%	92.4%
F_{CLN-4}	94.4%	90.5%	92.4%
F_{LDA}	93.8%	90.0%	91.9%

Table 4.9: Improvements of F_1-scores for Datasets 5 and 6.

	5	6
ΔF_{CON-1}	$-0.36 \pm 1.02\%$	$-0.71 \pm 1.20\%$
ΔF_{CON-2}	$-0.54 \pm 1.14\%$	$-1.13 \pm 1.51\%$
ΔF_{SVM-1}	$0.60 \pm 1.30\%$	$1.54 \pm 1.35\%$
ΔF_{SVM-2}	$0.48 \pm 1.27\%$	$1.35 \pm 1.30\%$
ΔF_{SVM-3}	$0.79 \pm 0.65\%$	$1.20 \pm 1.35\%$
ΔF_{SVM-4}	$0.80 \pm 0.63\%$	$1.14 \pm 1.31\%$
ΔF_{CLN-1}	$1.20 \pm 0.72\%$	$1.29 \pm 1.26\%$
ΔF_{CLN-2}	$1.20 \pm 0.66\%$	$1.39 \pm 1.26\%$
ΔF_{CLN-3}	$1.05 \pm 0.52\%$	$1.25 \pm 1.36\%$
ΔF_{CLN-4}	$1.08 \pm 0.52\%$	$1.28 \pm 1.32\%$
ΔF_{LDA}	$1.08 \pm 0.71\%$	$0.79 \pm 0.89\%$

CHAPTER 5

CONCLUSION

This thesis has explored algorithms for multimodal document classification unified in a meta-classification framework. We defined a framework that can be extended in a various ways, such as SVM-based meta-classification, Adaboost meta-classifier combination, image cleaning with text information, and LDA-based image meta-feature extraction. As shown in experiments on real-world multimodal documents from Wikipedia, meta-classification schemes that jointly exploit correlated or complementary information from both modalities generally improve the F_1-score. These schemes include SVM-based meta-classification, Adaboost for meta-classifier combination, and image cleaning with text information. Improvements are also observed in extensions such as LDA-based image meta-feature extraction and image meta-feature vector combination, which exploit the structure of multimodal documents and correlation within a modality.

REFERENCES

[1] C. H. Caldas and L. Soibelman, "Automating hierarchical document classification for construction management information systems," *Automation in Construction*, vol. 12, no. 4, pp. 395–406, Jul. 2003.

[2] E. Wiener, J. O. Pedersen, and A. S. Weigend, "A neural network approach to topic spotting," in *Proceedings 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995, pp. 317–332.

[3] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, vol. 29, no. 1, pp. 63–92, Oct. 2006.

[4] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM Computing Surveys*, vol. 41, pp. 12:1–12:31, Feb. 2009.

[5] J. Fagan, "Automatic phrase indexing for document retrieval," in *Proceedings of the 10th Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, 1987, pp. 91–101.

[6] Y. Yang and C. G. Chute, "Words or concepts: the features of indexing units and their optimal use in information retrieval," in *Proceedings of 17th Annual Symposium on Computer Applications in Medical Care*, 1993, pp. 685–689.

[7] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning*, 1997, pp. 412–420.

[8] G. Salton, "Developments in automatic text retrieval," *Science*, vol. 253, no. 5023, pp. 974–979, Aug. 1991.

[9] D. D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task," in *Proceedings of the 15th Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, 1992, pp. 37–50.

[10] T. Joachims, *Learning to Classify Text Using Support Vector Machines*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[11] R. E. Mayer and R. Moreno, "Aids to computer-based multimedia and learning," *Learning and Instruction*, vol. 12, no. 1, pp. 107–119, Feb. 2002.

[12] Y. Wang, Z. Liu, and J. C. Huang, "Multimedia content analysis using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, Nov. 2000.

[13] K. Lee and M. Cremer, "Segmentation-based lyrics-audio alignment using dynamic programming," in *Proceedings of the 9th International Conference of Music Information Retrieval*, 2008, pp. 395–400.

[14] H. Fujihara, M. Goto, J. Ogata, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic synchronization between lyrics and music cd recordings based on viterbi alignment of segregated vocal signals," in *Proceedings of the 8th IEEE International Symposium on Multimedia*, 2006, pp. 257–264.

[15] A. Quattoni, M. Collins, and T. Darrell, "Learning visual representations using images with captions," in *Proceedings of the 20th IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[16] W.-H. Lin and A. Hauptmann, "News video classification using svm-based multimodal classifiers and combination strategies," in *Proceedings of the 10th ACM International Conference on Multimedia*, 2002, pp. 323–326.

[17] W.-H. Lin and A. Hauptmann, "A meta-classification of multimedia classifiers," in *International Workshop on Knowledge Discovery in Multimedia and Complex Data*, 2002.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2003.

[20] "Rubik's cube," Oct. 2008. [Online]. Available: http://schools-wikipedia.org/wp/r/Rubik%2527s_Cube.htm

[21] "Text to matrix generator," Dec. 2008. [Online]. Available: http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG/

[22] M. Berry and M. Browne, *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Philadelphia, PA, USA: SIAM, 2005.

[23] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision*, 2004, pp. 1–22.

[24] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, pp. 264–271.

[25] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, Jun. 2001.

[26] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, vol. 21, no. 2, pp. 224–270, 1994.

[27] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proceedings of the European Conference on Computer Vision*, 2002, pp. 128–142.

[28] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Proceedings of the European Conference on Computer Vision*, 2006, pp. 128–142.

[29] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," in *Conference on Computer Vision and Pattern Recognition Workshop*, 2006, pp. 213–238.

[30] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[31] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[32] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the 20th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[33] V. N. Vapnik, *The Nature of Statistical Learning Theory.* New York, NY, USA: Springer, 1995.

[34] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, June 1998.

[35] S. Abe, *Support Vector Machines for Pattern Classification*. London, UK: Springer, 2005.

[36] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: A stepwise procedure for building and training a neural network," in *Neurocomputing: Algorithms, Architectures and Applications*. New York, NY, USA: Springer-Verlag, 1990, vol. F68 of NATO ASI Series, pp. 41–50.

[37] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[38] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Journal of the Royal Statistical Society: Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[39] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," *Bioinformatics*, vol. 1, no. 1, pp. 1–16, 2003.

[40] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[41] Y. Freund and R. E. Schapire, "A short introduction to boosting," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999, pp. 1401–1406.

[42] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 1998.

[43] S. Chen, V. Monga, and P. Moulin, "Meta-classifiers for multimodal document classification," in *IEEE International Workshop on Multimedia Signal Processing*, 2009, pp. 1–6.

[44] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Engineering Bulletin*, vol. 23, pp. 3–13, 2000.

[45] A. Esuli and F. Sebastiani, "Training data cleaning for text classification," in *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, 2009, pp. 29–41.

[46] T. Okita, "Data cleaning for word alignment," in *Proceedings of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing 2009 Student Research Workshop*, 2009, pp. 72–80.

[47] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[48] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[49] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, 1999, pp. 50–57.

[50] K. Min, Z. Zhang, J. Wright, and Y. Ma, "Decomposing background topics from keywords by principal component pursuit," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010, pp. 269–278.

[51] X. Wei and W. B. Croft, "Lda-based document models for ad-hoc retrieval," in *Proceedings of the 29th Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, 2006, pp. 178–185.

[52] L. Fei-fei, "A bayesian hierarchical model for learning natural scene categories," in *Conference on Computer Vision and Pattern Recognition Workshop*, 2005, pp. 524–531.

[53] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," in *Proceedings of the 10th IEEE International Conference on Computer Vision*, 2005, pp. 370–377.

[54] "Taipei," Oct. 2008. [Online]. Available: http://schools-wikipedia.org/wp/t/Taipei.htm

[55] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 331–339.

[56] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[57] "2008/9 wikipedia selection for schools," Oct. 2008. [Online]. Available: http://schools-wikipedia.org

[58] "Clarinet," Oct. 2008. [Online]. Available: http://schools-wikipedia.org/wp/c/Clarinet.htm